FILE NO. AP31508

BAKER BOTTS, L.L.P.

30 ROCKEFELLER PLAZA

NEW YORK, NEW YORK 10112

TO ALL WHOM IT MAY CONCERN:

Be it known that WE, ANTHONY DAVID PEACHMAN and IAN STEPHEN SIMMONS, citizens of GREAT BRITAIN, whose post office addresses are 31 Ashdown Avenue, Saltdean, Brighton, East Sussex BN2 8AH, and The Elms, School Road, Broughton, Cambs PE17 3AT, respectively, have invented an improvement in

CONFIGURATION OF IC CARD

of which the following is a

SPECIFICATION

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of United States Patent Application Serial No. 09/162,605 filed on September 29, 1998, entitled "Configuration of IC Card," which claims priority to United States Provisional Application Serial No. 60/073,906 filed on February 6, 1998, entitled "Remote Configuration of IC Card," and United States Provisional Application Serial No. 60/072,561, filed on January 22, 1998, entitled "Codelets," all of which are incorporated herein by reference.

RELATED APPLICATIONS

[0002] This application is related to U.S. Patent Application Serial No. 09/076,551 filed on May 12, 1998 entitled "Secure Multiple Application Card System and Process," which is incorporated herein by reference.

BACKGROUND OF INVENTION

[0003] Integrated circuit cards are becoming increasingly used for many different purposes in the world today. An IC card typically is the size of a conventional credit card on which a computer chip is embedded. It comprises a microprocessor, read-only-memory (ROM), electrically erasable programmable read-only-memory (EEPROM), an Input/Output (I/O) mechanism and other circuitry to support the microprocessor in its operations. An IC card may contain one or more applications in memory. MULTOS™ is a multiple application operating system which runs on IC cards, among other platforms, and allows multiple applications to be executed on the card itself. This allows a card user to run many programs stored in the card (for example, credit/debit, electronic money/purse and/or loyalty applications) irrespective of the type of terminal (i.e., ATM, telephone and/or POS) in which the card is inserted for use.

[0004] IC cards typically have limited storage capacity due to the size and cost restraints of locating memory on the card. Applications for multi-application smart cards are written in a programming language and are typically stored in the EEPROM whose contents can be changed during the lifetime of the card. One example of a programming language used in IC cards is the

Multos Executable Language (MEL™). The MEL program instructions are read from EEPROM when they are executed and are interpreted by the operating system stored in ROM.

[0005] The ROM on the IC card includes the operating system written in assembler language code for the particular integrated circuit configuration (native language type code). The operating system code stored in ROM is fixed when the ROM is initially written and the information stored in ROM will not change for the life of the card.

[0006] Also present in ROM can be subroutines called primitives written in a native language code for the microprocessor which can be called by either the operating system itself or by applications when they are executed. Primitives are written in native language (i.e. assembler language) so that they can be executed very quickly and minimal interpretation of the instructions is necessary for execution. These primitives are collections of instructions which typically perform a desired function, such as a mathematical or cryptographic function. The instructions are never changed during the lifetime of the card. Any data used or accessed by the primitives are stored in EEPROM so that the contents of the data elements can change as necessary.

[0007] Also capable of being stored in ROM are "codelets," which are sets of instructions written in a programming language (not native language code). These codelets can be stored in ROM so as to maximize the usage of memory and allow ROM to store complete applications as well as primitives. The codelet can be as small as one instruction or as large as will fit into the remaining ROM memory space. For example, the purse application described above can be stored in ROM when the card is initialized in order to free up space in EEPROM for additional applications which can be loaded at any time.

[0008] Once data is stored in ROM, the data can never be modified or deleted and new data cannot be added after ROM is set. Moreover, in prior art systems, when the chip card is manufactured, a primitive address table is stored on the card which allows the operating system to locate the memory address of a primitive. This address table in ROM is also permanently set.

[0009] In this system, described in an application copending with this one (see Serial No. 09/076,551, incorporated herein by reference), subsequent to card manufacture (at which time the ROM is fixed), the card is "personalized." This personalization step takes place either shortly after the card is made or anytime thereafter, up to a period of months or more. In the meantime, before the card is personalized, cards remain "blank" (i.e., unassigned to an individual user or group) and typically will be held at the card manufacturer or card issuer until needed. During this stage, because the cards have not yet been personalized, there is a greater risk that the cards would be improperly used.

[00010] The personalization step -- in which the cards are assigned to a particular user or group -- takes place at a location different from the card manufacturer generally under control of the card-issuer (i.e., the bank issuing the card) or some other personalization bureau ("PB"). A separate and preferably centrally located Certification Authority, which oversees the cards' interaction, provides the usually remote PB with appropriate security data, discussed below, to allow the PB to personalize (i.e., enable) the card, and to allow an application provider to load (either at the time of enablement or later) an application program, such as a purse application, onto the card.

[00011] One of the problems confronting multi-application card designers is how to address the situation where after the primitive or codelet is masked or otherwise stored in the

ROM at the time of manufacture (and thus cannot thereafter be changed), the primitive or codelet needs to be replaced, modified or updated to fix a bug or to take advantage of a more efficient or effective routine. Another concern is to insure that the original primitives and codelets masked into ROM are not capable of use until the card is personalized, i.e. enabled for a particular user or group, with individual keys and identifiers. Accordingly it is an object of the invention to provide a method and system which solves these problems.

SUMMARY OF THE INVENTION

[00012] The applicant here has determined that one way to achieve these objectives is by loading in EEPROM at the personalization step and not at the manufacturing step an address table assigning to each primitive and/or codelet a name and corresponding address identifying where the primitive and/or codelet can be found in memory. In this manner, if the primitive masked in ROM at the time of manufacture needs to be changed, only the address for that primitive needs to be changed to point to the location in which the updated primitive sits. In addition, since neither an application nor the operating system will know where the primitive is located without a stored address table, the primitives cannot be called and the card cannot run until the primitive address table is loaded at the personalization step. This prevents use of the card until it is enabled at the personalization step.

BRIEF DESCRIPTION OF THE DRAWINGS

[00013] Further objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying figures showing illustrative embodiments of the invention, in which

[00014] Fig. 1 is a block diagram illustrating the three states in the life of a multi-application IC card in a secure system;

[00015] Fig. 2 is a block diagram showing the components of the system architecture for the enablement process of an IC card in a secure multi-application IC card system;

[00016] Fig. 3 is a block diagram illustrating the read only memory space segments for an IC card at the time of manufacture in accordance with the present invention;

[00017] Fig. 4 is a block diagram illustrating the electrically erasable programmable read-only-memory space segments for an IC card after it has been loaded at the personalization stage in accordance with the present invention;

[00018] Fig. 5 is a block diagram illustrating the address table loaded in EEPROM of an IC card at the personalization stage, in accordance with the present invention;

[00019] Fig. 6 illustrates an integrated circuit card which can be used in connection with this invention; and

[00020] Fig. 7 is a functional block diagram of the integrated circuit shown in Fig. 6.

[00021] Throughout the figures, the same reference numerals and characters, unless otherwise stated, are used to denote like features, elements, components or portions of the illustrated embodiments. Moreover, while the subject invention will now be described in detail

with reference to the figures, it is done so in connection with the illustrative embodiments. It is intended that changes and modifications can be made to the described embodiments without departing from the true scope and spirit of the subject invention as defined by the appended claims.

DETAILED DESCRIPTION OF THE DRAWINGS

[00022] Figure 1 shows the three steps involved in providing an operational multi-application IC card in a secure system. The first step is the card manufacturing step 101. The second step is the personalization step 103 where card personalization data (also called entity authentication data) is loaded onto the card. The third step is the application loading step 105 which checks to see if a card is qualified to receive an application, i.e., when the personalization data is checked against the application permissions data associated with the application to be loaded. Each of these three steps is described in detail in co-pending application serial number 09/076,551.

[00023] Figure 2 shows the components of the system architecture for the card initialization process of an IC card in a secure multiple application IC card system. The system includes a card manufacturer 102, a personalization bureau 104, an application loader 106, the IC card 107 being initialized, the card user 109 and the certification authority 111 for the entire multiple application secure system. The card user 109 is the person or entity who will use the stored applications on the IC card.

[00024] The card user would contact a card issuer 113, such as a bank which distributes IC cards, and request an IC card with the two applications both residing in memory of a single IC

card. The integrated circuit chip for the IC card would be manufactured by manufacturer 102 and sent to the card issuer 113 (or an entity acting on its behalf) in the form of an IC chip on a card. During the manufacturing process, data is transmitted 115 via a data conduit from the manufacturer 102 to card 107 and stored in IC card 107's memory. (Any of the data conduits described in this figure could be a telephone line, Internet connection or any other transmission medium.) The certification authority 111, which maintains encryption/decryption keys for the entire system, transmits 117 security data (i.e., global public key) to the manufacturer over a data conduit which is placed on the card by the manufacturer along with other data, such as the card enablement key and card identifier. The card's multiple application operating system is also stored in ROM and placed on the card by the manufacturer. After the cards have been initially processed, they are sent to the card issuer for personalization and application loading.

[00025] The card issuer 113 performs, or has performed by another entity, two separate functions. First, the personalization bureau 104 personalizes the IC card 107 in the ways described above, and second, the application loader 106 loads the application provided the card is qualified, as described in copending application serial number 09/076,551.

[00026] Backtracking now to the time of manufacture, the ROM 120 of IC card is loaded, as illustrated in Figure 3, with operating system code 122, codelets 1 and 2 identified respectively as 124, 126, at addresses 1000 and 1050, and primitives 1, 2, 3, 4 identified respectively are 128, 130, 132, 134, at addresses 2020, 2040, 2080, and 3000. The addresses are preferably physical addresses in ROM, an offset from a primitive starting pointer, or any other addressing scheme.

[00027] Subsequently, as described above, the card is personalized. The CA provides the PB with personalization information, which may include an individual key set 136. This information is sent to the PB usually at a remote location either through the Internet, by CD ROM or other data conduit or storage device. The PB remotely loads this information onto the EEPROM of the card (see Fig. 4) along with certain identifiers 138, such as a card identification, an issuer identification, product type identification (representing the type of application, i.e., purse, loyalty, etc.) and the date of loading. Additional primitive or codelet code can also be loaded at this time.

[00028] In accordance with this invention, the PB further remotely loads onto the EEPROM of the card the codelet/primitive address table 140. As shown in Fig. 5, this address table 140 contains a listing of the names of the codelets and primitives to be called by either the application program or operating system together with the memory addresses containing the code to be called. The location of code corresponding to a primitive call by the operating system or an application will be determined at this time. Thus, the controlling authority or system operator can select which version of code stored in the card will be executed when a particular primitive name is called.

[00029] In this particular case, a program instruction such as:

CALL PRIM 4 (DATA)

would result in a search of the address table to locate the address of PRIM 4. Because a new PRIM 4, with address 3080, was added into the programmable portion of the card memory at time of personalization to replace old PRIM 4, the operating system will simply fetch the new PRIM 4 at location 3080 as indicated in the address table. The old code at memory location

3000 will never be accessed by the operating system because there is no entry in the address table pointing to the old code.

[00030] Accordingly, this remote loading of an address table at the time of personalization allows the system (1) to control enablement until desired; and (2) to make use of a card despite an outdated codelet or primitive which may have been permanently placed in the card at the time of manufacture.

[00031] Figure 6 illustrates a card 600 incorporating integrated circuit technology that can be used with the presently claimed invention. Card 600 looks similar to a conventional credit card, but also includes integrated circuit (IC) 622, which contains a microprocessor, and electrical contacts 624 for communication between IC 622 and devices external to card 600. Card 600 can be used for example, as a credit card, a debit card, and\or as an electronic cash card, i.e., a card containing monetary value that can be transferred when the cardholder makes purchases, for example, a MONDEX™ cash card.

[00032] Figure 7 is a functional block diagram of the IC section 622 and contains at least processing unit 710 and memory unit 750. Preferably, IC 722 also includes control logic 720, a timer 730, and input/output ports 740. IC section 722 can also include a co-processor 760. Control logic 720 provides, in conjunction with processing unit 710, the control necessary to handle communications between memory unit 750 and input/output ports 740. Timer 730 provides a timing reference signal for processing unit 710 and control logic 720. Co-processor 760 provides the ability to perform complex computations in real time, such as those required by cryptographic algorithms.

[00033] The foregoing merely illustrates the principles of the invention. It will thus be appreciated that those skilled in the art will be able to devise numerous systems and methods which, although not explicitly shown or described herein, embody the principles of the invention and are thus within the spirit and scope of the invention.